

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И МАШИННОЕ ОБУЧЕНИЕ/ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DOI: https://doi.org/10.60797/IRJ.2026.169.21 EDN: WGLDRS

ДОСТУП К ОБЪЕКТАМ SVG-СЦЕН НА УРОВНЕ ОБЪЕКТОВ: БЕНЧМАРК И РАСТРОВО-ОБОСНОВАННЫЙ МЕТОД ИЗВЛЕЧЕНИЯ ВЕКТОРНЫХ ОБЪЕКТОВ ПО ТЕКСТОВОМУ ЗАПРОСУ

Научная статья

Тимофеенко Б.А.[1,] *, Ефимова В.А.[2], Жарский И.А.[3], Малашенко Б.Т.[4]
[1] ORCID : 0009-0009-6281-7239;
[4] ORCID : 0009-0001-8553-8883;
[1, 2, 3, 4] Университет ИТМО, Санкт-Петербург, Российская Федерация

* Корреспондирующий автор (boriswinner88[at]gmail.com)

Предложена: 30.05.2026; Принята: 23.06.2026; Опубликована: 17.07.2026

Аннотация

SVG-графика представляет собой редактируемые структурированные документы, однако семантические объекты в SVG-сценах часто не совпадают с контурами (paths), группами или другими элементами дерева документа, что затрудняет доступ на уровне объектов в творческих рабочих процессах, требующих извлечения, редактирования или повторного использования частей векторных ресурсов. В работе вводится задача извлечения векторных объектов по текстовому запросу, цель которой — восстановить редактируемый под-SVG, соответствующий запросу на естественном языке. Предложен бенчмарк с аннотациями на уровне узлов и оценкой, учитывающей векторную структуру, содержащий 2793 оценочных образца. Также предложен растрово-обоснованный метод извлечения, который локализует запрашиваемый объект на отрендеренном изображении и сопоставляет полученную область с примитивами SVG посредством фильтрации на основе перекрытия. Эксперименты показывают, что метод обеспечивает сильную базовую линию и превосходит структурированные базовые методы на основе больших языковых моделей по метрикам перекрытия на уровне узлов. Результаты подчёркивают необходимость в моделях и протоколах оценки, учитывающих специфику SVG, для редактируемой векторной графики на уровне объектов.

Ключевые слова: векторная графика, SVG, извлечение объектов по текстовому запросу, доступ на уровне объектов, бенчмарк, растровое обоснование, открытословарная локализация, сегментация изображений, Grounding DINO, Segment Anything Model.

OBJECT-LEVEL ACCESS TO SVG SCENE OBJECTS: A BENCHMARK AND A RASTER-BASED METHOD FOR EXTRACTING VECTOR OBJECTS FROM A TEXTUAL QUERY

Research article

Timofeenko B.A.[1,] *, Efimova V.A.[2], Zharskii I.A.[3], Malashenko B.T.[4]
[1] ORCID : 0009-0009-6281-7239;
[4] ORCID : 0009-0001-8553-8883;
[1, 2, 3, 4] ITMO National Research University, Saint-Petersburg, Russian Federation

* Corresponding author (boriswinner88[at]gmail.com)

Suggested: 30.05.2026; Accepted: 23.06.2026; Published: 17.07.2026

Abstract

SVG graphics are edible structured documents; however, semantic objects in SVG scenes often do not correspond to paths, groups or other elements of the document tree, which makes it difficult to access them at the object level in creative workflows that require the extraction, editing or reuse of parts of vector resources. The work introduces the task of extracting vector objects based on a text query, with the aim of reconstructing an editable sub-SVG corresponding to the natural-language query. A benchmark is suggested with node-level annotations and a scoring scheme that takes the vector structure into account, containing 2,793 evaluation samples. A raster-based extraction method is also proposed, which localises the requested object in the rendered image and maps the resulting region to SVG primitives via overlap-based filtering. Experiments show that the method provides a strong baseline and outperforms structured baseline methods based on large language models on node-level overlap metrics. The results highlight the need for models and evaluation protocols that take into account the specific characteristics of SVG for editable vector graphics at the object level.

Keywords: vector graphics, SVG, object extraction via text queries, object-level access, benchmark, raster grounding, open-vocabulary localisation, image segmentation, Grounding DINO, Segment Anything Model.

Введение

Векторная графика приобретает всё большее значение для творческих ИИ-систем, в которых визуальный контент должен не только генерироваться, но и анализироваться, редактироваться, повторно использоваться и комбинироваться [1], [2], [5], [6]. Современные бенчмарки для SVG также рассматривают SVG как структурированную модальность для понимания, редактирования и мультимодального рассуждения [8], [9], [10], [11]. Среди векторных форматов SVG особенно привлекателен тем, что представляет изображение как структурированный документ, состоящий из

геометрических примитивов, групп, стилей, преобразований и переиспользуемых определений. Эта структура делает SVG пригодным для рабочих процессов, в которых пользователи взаимодействуют с визуальным контентом на более тонком уровне, чем изображение целиком.

Однако во многих творческих сценариях пользователям необходим доступ к существующей сцене на уровне объектов. Например, дизайнер может захотеть извлечь окно из архитектурной иллюстрации, перекрасить дерево в наборе иконок или повторно использовать все облака из декоративной SVG-сцены. Такие операции естественны для пользователей-людей, но они не поддерживаются напрямую исходной структурой SVG-документа. Семантический объект может быть распределён по нескольким контурам, тогда как одна группа SVG может содержать визуально не связанные между собой части. Стили, преобразования, обтравочные контуры и ссылочные определения ещё больше усложняют связь между визуальными объектами и XML-узлами. В результате редактируемая структура, доступная в файле, часто не соответствует объектной структуре, необходимой в творческих рабочих процессах.

Этот разрыв мотивирует задачу извлечения векторных объектов по текстовому запросу. По заданным SVG-сцене и запросу на естественном языке цель состоит в том, чтобы восстановить указанный объект как самостоятельный редактируемый под-SVG. Задача отличается от детекции или сегментации растровых объектов, поскольку ограничивающая рамка или пиксельная маска не идентифицируют векторные примитивы, которые должны быть сохранены. Она также отличается от генерации SVG, поскольку выходные данные должны собираться из элементов, уже присутствующих во входной сцене, а не синтезироваться как новый рисунок. Насколько известно авторам, эта форма доступа к объектам SVG-сцен по текстовому запросу не рассматривалась напрямую существующими бенчмарками.

Поставленная задача решается с помощью бенчмарка и растрово-обоснованного метода извлечения. Бенчмарк предоставляет аннотации на уровне узлов, позволяя оценить, выбирает ли метод правильные редактируемые векторные элементы. Предложенный метод связывает растровое семантическое обоснование с векторным редактированием: он рендерит SVG-сцену, локализует запрашиваемый объект с помощью модели Grounding DINO [12], при необходимости уточняет область с помощью модели SAM [13] и сопоставляет полученную область с примитивами SVG посредством фильтрации на основе перекрытия. Затем выбранные примитивы реконструируются в самостоятельный под-SVG, который можно повторно использовать или редактировать независимо.

Предложенный метод дополнительно сравнивается со структурированными базовыми методами на основе больших языковых моделей (LLM), которые напрямую предсказывают маски узлов SVG. Это сравнение актуально, поскольку LLM всё чаще используются как универсальные контроллеры для кода и визуальных ресурсов. Результаты показывают, что даже в условиях структурированного предсказания современные LLM остаются ограниченными для точного извлечения SVG-объектов, особенно для длинных или сложных SVG-документов, тогда как фильтрация примитивов на основе растрового обоснования обеспечивает сильный и легковесный подход к восстановлению редактируемых объектов из SVG-сцен.

Основной вклад работы состоит в следующем. Во-первых, вводится новая задача — извлечение векторных объектов по текстовому запросу — для восстановления редактируемых объектов из существующих SVG-сцен. Во-вторых, предлагается бенчмарк с аннотациями на уровне узлов и оценкой, учитывающей векторную структуру. В-третьих, предлагается растрово-обоснованный метод извлечения, сопоставляющий локализацию, обусловленную текстом, с редактируемыми примитивами SVG. В-четвёртых, предложенный метод оценивается в сравнении со структурированными базовыми методами на основе LLM, и анализируются практические ограничения SVG-рассуждений на основе LLM для точного доступа на уровне объектов.

Методы и принципы исследования

2.1. Обзор связанных работ

Ранние работы, такие как DeepSVG [1], изучают обучаемые представления и генерацию SVG-иконок, тогда как методы дифференцируемой растеризации, такие как DiffVG [2], связывают векторную графику с оптимизацией в растровом пространстве. Более поздние методы генерации векторов, обусловленной текстом, синтезируют SVG или векторные эскизы из подсказок, используя руководство «язык–изображение», диффузионные приоры или авторегрессионное моделирование [3], [4], [7]. Параллельно бенчмарки, такие как SVGenius [9], VectorGym [10] и SVGEditBench V2 [11], оценивают понимание, редактирование, генерацию и манипуляцию кодом SVG в широком диапазоне задач. В совокупности эти работы устанавливают SVG как важную область для структурированного визуального рассуждения и творческой манипуляции. Однако существующие бенчмарки в основном сосредоточены на генерации, редактировании, описании или преобразовании растра в SVG, а не на восстановлении редактируемых семантических объектов из уже существующей SVG-сцены.

Смежное направление работ изучает семантическую организацию в векторной графике. SemLayer [14] исследует восстановление семантических слоёв из плоских векторных ресурсов, тогда как AmodalSVG [15] строит семантически организованные векторные слои для последующего редактирования. Эти работы показывают, что символическая структура реальных SVG-ресурсов часто не совпадает с границами семантических объектов. Настоящая работа мотивирована тем же наблюдением, но решает другую задачу: вместо семантической реконструкции или послойной векторизации изучается извлечение редактируемых объектов из существующих SVG-сцен, обусловленное запросом.

Наиболее близкой смежной задачей извлечения является WildSVG [16], изучающая извлечение SVG из растровых и естественных изображений. В отличие от неё, рассматриваемая постановка исходит из существующего SVG-документа и спрашивает, какие векторные примитивы соответствуют текстовому запросу. В более широком смысле растровое обоснование обеспечивает естественный интерфейс для анализа SVG-сцен с помощью современных моделей зрения. Grounding DINO [12] обеспечивает локализацию с открытым словарём из текстовых подсказок, SAM [13] поддерживает сегментацию по подсказке, а Grounded SAM [17] объединяет эти компоненты в практичный

конвейер локализации, обусловленной текстом. Однако эти методы работают в растровом пространстве и не восстанавливают напрямую редактируемые примитивы SVG. Насколько известно авторам, ни один существующий бенчмарк или метод не решает напрямую задачу извлечения векторных объектов из существующих SVG-сцен по текстовому запросу.

2.2. Задача и бенчмарк

2.2.1. Определение задачи

Извлечение векторных объектов по текстовому запросу вводится как задача доступа к существующим SVG-сценам на уровне объектов. По заданным SVG-документу и запросу на естественном языке цель состоит в том, чтобы восстановить указанный объект или набор объектов как самостоятельный редактируемый под-SVG. На структурном уровне задачу можно рассматривать как фильтрацию элементов SVG, обусловленную запросом: из полного SVG-документа метод должен выбрать элементы, принадлежащие запрашиваемому визуальному содержанию.

Запрос может относиться к одному экземпляру объекта или к нескольким объектам в одной сцене. Например, «солнце» может соответствовать одному объекту, тогда как «облака», «глаза» или «окна» могут соответствовать нескольким объектам. Соответственно, метод может возвращать список выборов объектов для одного запроса, причём список содержит либо одну маску, либо несколько. Это аналогично детекции и сегментации с открытым словарём, где запрос категории может давать несколько детекций, но здесь выходные данные в конечном итоге должны быть выражены в терминах элементов SVG, а не пикселей.

Как целевые объекты, так и предсказания метода представляются в виде бинарных масок над узлами SVG. Каждый SVG разбирается как XML-дерево с использованием детерминированного порядка узлов. Положительный узел указывает, что соответствующий элемент SVG принадлежит запрашиваемому объекту или набору объектов. Целевые узлы переднего плана соответствуют отрисовываемым примитивам SVG, включая path, rect, circle, ellipse, line, polyline и polygon. Структурные элементы, такие как группы, не аннотируются как целевые примитивы напрямую, однако остаются важными для создания корректного под-SVG, поскольку выбранные отрисовываемые примитивы должны быть реконструированы вместе с необходимыми предками и ссылочными определениями.

Эта формулировка отличается от детекции и сегментации растровых объектов. Растровая ограничивающая рамка или пиксельная маска могут указать, где объект появляется на отрендеренном изображении, но не определяют, какие векторные примитивы должны быть сохранены для редактирования. Задача также отличается от генерации SVG, поскольку выходные данные собираются из существующих элементов входного документа, а не синтезируются как новый векторный рисунок. Таким образом, извлечение векторных объектов по текстовому запросу специально оценивает доступ к внутренней структуре существующей SVG-сцены на уровне объектов.

2.2.2. Построение бенчмарка

Предложен бенчмарк для извлечения векторных объектов из SVG-сцен по текстовому запросу. Бенчмарк содержит 268 SVG-сцен и 2271 корректную аннотацию объектов после исключения отклонённых записей. Каждая аннотация содержит SVG-сцену, один или несколько текстовых запросов и эталонную бинарную маску над узлами SVG. Несколько текстовых запросов могут описывать один и тот же аннотированный объект; например, короткое название объекта и более конкретную фразу. После раскрытия таких вариантов запросов и разрешения случаев повторяющихся объектов бенчмарк содержит 2793 оценённые пары SVG–запрос.

Бенчмарк спроектирован как оценочный набор данных, а не как обучающий. Он содержит реалистичные структуры SVG-документов, включая группы, определения, обтравочные контуры, преобразования и атрибуты стилей. Эти структуры важны, поскольку семантические объекты в SVG-сценах часто не совпадают с существующими группами или простыми XML-поддеревьями. Сводная статистика бенчмарка приведена в таблице 1.

Таблица 1 - Сводная статистика предложенного бенчмарка для извлечения векторных объектов по текстовому запросу

DOI: https://doi.org/10.60797/IRJ.2026.169.21.1

Показатель	Значение
SVG-сцены	268
Корректные аннотации	2271
Оценочные образцы	2793
Уникальные текстовые запросы	1565
Среднее число XML-узлов на SVG	84,4
Медианное число XML-узлов на SVG	45
Макс. число XML-узлов на SVG	1362
Среднее число отрисовываемых примитивов на SVG	61,6

2.2.3. Типы аннотаций

Бенчмарк включает три типа аннотаций, отражающих различные формы доступа к объектам SVG-сцен. Первый тип — один видимый объект — соответствует запросам, относящимся к одному видимому объекту в сцене (например,

«солнце», «телефон» или «дерево»); в этом случае целевая маска содержит элементы SVG, образующие этот единственный объект. Второй тип — несколько видимых объектов — соответствует запросам, естественно относящимся к нескольким видимым объектам или повторяющимся семантическим частям (например, «облака», «глаза» или «окна»); для таких аннотаций целевая маска включает все соответствующие объекты в сцене, что важно для рабочих процессов, где пользователь редактирует или повторно использует целый семантический набор, а не один изолированный экземпляр.

Третий тип — повторяющиеся экземпляры объектов — рассматривает случаи, когда сцена содержит несколько похожих экземпляров, и один лишь текстовый запрос не идентифицирует конкретный из них. Например, если сцена содержит несколько окон, а запрос — это просто «окно», выбор одного произвольного экземпляра сделал бы оценку неоднозначной. Для таких случаев аннотации повторяющихся экземпляров группируются по SVG-сцене и запросу во время оценки, а целевая маска определяется как объединение всех соответствующих масок экземпляров. Этот протокол делает задачу корректно определённой при запросах только из текста, сохраняя при этом объектную природу бенчмарка.

2.3. Протокол оценки

Бенчмарк оценивает методы на уровне выбора узлов SVG. Для каждой пары SVG–запрос метод возвращает список предсказанных выборов объектов. Каждый выбор может происходить из обнаруженной растровой области, маски сегментации или прямого предсказания над узлами SVG. Независимо от метода, каждое предсказание преобразуется в бинарную маску над узлами входного SVG. Поскольку один запрос может соответствовать нескольким объектам, все предсказанные выборы для одного запроса объединяются в одну совмещённую маску узлов, которая сравнивается с эталонной совмещённой маской для соответствующего набора объектов. Таким образом, оценка измеряет, восстанавливает ли метод полное редактируемое содержание SVG, связанное с запросом. Сопоставление по экземплярам между предсказанными и эталонными объектами не выполняется, поскольку оценка на уровне экземпляров для наборов узлов SVG требует дополнительных правил разрешения неоднозначности и оставлена для будущей работы.

2.3.1. Метрики на уровне узлов

Основные метрики сравнивают предсказанные и эталонные наборы выбранных узлов SVG: приводятся IoU на уровне узлов (Node IoU), F1-мера (Node F1) и точное совпадение. Node IoU измеряет перекрытие между предсказанными и эталонными положительными узлами; F1 уравновешивает пропуск целевых узлов и выбор лишних узлов; точное совпадение равно единице только тогда, когда предсказанная маска узлов идентична эталонной. Эти метрики являются центральными, поскольку они напрямую оценивают извлечение редактируемых векторов: метод, который локализует правильную растровую область, но выбирает неправильные примитивы SVG, получит низкую оценку на уровне узлов, и наоборот — визуально малый, но структурно важный примитив учитывается явно, если он является частью эталонного объекта.

2.3.2. Метрики на отрендеренных изображениях

Дополнительно оценивается отрендеренный вид извлечённых объектов. Для каждого образца предсказанная и эталонная маски узлов преобразуются в подмножества SVG и растеризуются, после чего вычисляются стандартные метрики уровня изображения между отрендеренным предсказанием и отрендеренным эталонным извлечением, включая MSE, MAE, PSNR и SSIM. Метрики на отрендеренных изображениях приводятся как вспомогательные, а не как основной критерий: они полезны для измерения визуального согласия, но не полностью отражают, были ли выбраны правильные редактируемые элементы SVG. Чтобы снизить влияние пустых фоновых областей, эти метрики дополнительно вычисляются на области интереса переднего плана, определяемой как объединение отрендеренного предсказания и эталона.

2.3.3. Агрегация

Если не указано иное, приводятся макроусреднённые метрики по оценённым парам SVG–запрос. Для метрик на уровне узлов также используются микроусреднённые оценки в сравнениях, где глобальные количества истинно положительных, ложно положительных и ложно отрицательных результатов информативны. При сравнении с базовыми методами на основе LLM результаты приводятся на подмножестве SVG-сцен, которые помещаются во входной контекст LLM, причём для всех методов используется один и тот же протокол оценки по совмещённой маске.

2.4. Метод

Предложенный метод сопоставляет растровую локализацию, обусловленную текстом, с редактируемой структурой SVG. По заданным SVG-сцене и запросу он сначала идентифицирует релевантные запросу области в растровом пространстве, а затем выбирает примитивы SVG, чья отрендеренная опора лежит внутри этих областей. Метод является модульным и производит список кандидатов на выбор объектов — по одному для каждой обнаруженной области или для объединённого набора областей. Общая схема конвейера показана на рисунке 1.

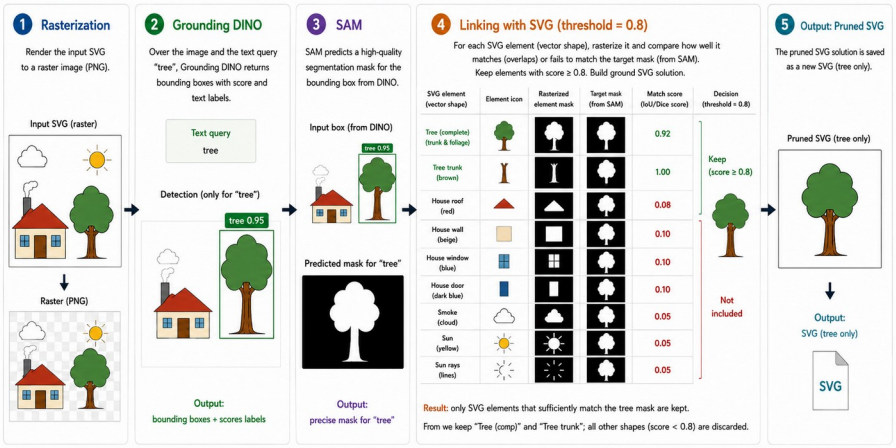

Рисунок 1 - Обзор предложенного растрово-обоснованного метода извлечения SVG-объектов: растеризация входного SVG, локализация запрашиваемого объекта моделью Grounding DINO, опциональное уточнение области моделью SAM, поэлементная оценка примитивов SVG относительно целевой маски и экспорт результирующего поддерева SVG

DOI: https://doi.org/10.60797/IRJ.2026.169.21.2

2.4.1. Общая схема конвейера

Конвейер состоит из четырёх этапов. Сначала входной SVG растеризуется в изображение, используемое для визуального обоснования. Во-вторых, Grounding DINO локализует области, соответствующие текстовому запросу. В-третьих, каждая обнаруженная рамка преобразуется в бинарную целевую маску — либо путём прямого заполнения рамки, либо путём использования рамки как подсказки для SAM. Наконец, отрисовываемые примитивы SVG рендерятся независимо, оцениваются относительно целевой маски и фильтруются согласно их перекрытию. Выбранные примитивы экспортируются вместе с необходимой структурой SVG для формирования редактируемого под-SVG.

2.4.2. Растровое обоснование

SVG-сцена растеризуется в фиксированный аналитический холст с сохранением геометрии сцены. Тот же холст используется для рендера полной сцены, входа детектора, целевых масок и рендеров отдельных примитивов, что делает возможным попиксельное сравнение растровых областей объектов с отрендеренными примитивами SVG. Для локализации с открытым словарём используется Grounding DINO [12]. Входной запрос — это короткая именная группа, такая как «дерево», «телефон» или «облака». Детектор возвращает одну или несколько ограничивающих рамок, соответствующих релевантным запросу областям; поскольку один запрос может относиться к нескольким объектам, метод сохраняет все обнаруженные области. Для каждой обнаруженной рамки строится целевая маска: в режиме «только рамка» это заполненная прямоугольная рамка, а в режиме сегментации рамка передаётся в SAM [13], который предсказывает уточнённую бинарную маску. Это даёт два варианта метода — легковесную версию с заполнением рамки и версию, уточнённую SAM.

2.4.3. Фильтрация на уровне примитивов

Ключевой шаг — преобразовать растровую целевую маску в выбор примитивов SVG. Перечисляются отрисовываемые листовые элементы дерева SVG: в реализации это path, rect, circle, ellipse, line, polyline и polygon; структурные узлы, такие как группы, напрямую не оцениваются. Для каждого отрисовываемого примитива рендерится мини-SVG, содержащий этот примитив в его исходном структурном контексте, что производит бинарную маску примитива на том же холсте, что и целевая маска. Рендеринг примитивов вместо ручного разбора их геометрии делает процедуру устойчивой к специфичным для SVG механизмам, таким как преобразования, наследуемые стили, обтравка и ссылочные определения. Для маски примитива C_i и целевой маски T вычисляется оценка перекрытия, учитывающая включение:

$$s_i = |C_i \cap T| / |C_i| \quad (1)$$

где C_i — маска i-го примитива, T — целевая маска, а $|\cdot|$ обозначает число активных пикселей. Если у примитива нет видимых пикселей, оценка полагается равной нулю. Оценка измеряет, какая доля примитива лежит внутри целевой области. Эта направленная оценка лучше подходит для фильтрации примитивов, чем симметричный IoU, поскольку малый примитив может полностью содержаться в большой маске объекта, имея при этом низкий симметричный IoU. Примитив сохраняется, если его оценка составляет не менее τ; во всех приведённых экспериментах используется τ = 0,8. Для каждой обнаруженной области это производит одного кандидата на выбор объекта; когда для одного запроса обнаруживается несколько областей, метод может либо сохранять их как отдельные выборы, либо объединять их целевые маски для извлечения полного набора запрашиваемых объектов.

2.4.4. Реконструкция SVG

После фильтрации примитивов сохранённые отрисовываемые элементы помечаются в исходном дереве SVG. Для сохранения корректного рендеринга также сохраняются необходимые предки и ссылочные определения, включая структурные элементы и записи в defs, необходимые для стилей, преобразований, обтравки или иных зависимостей

рендеринга. Поэтому выходные данные — это не растровая вырезка и не плоский список контуров, а структурно корректный под-SVG, собранный из исходного документа. Извлечённый SVG можно редактировать как векторную графику, а также растеризовать для визуализации или для оценки на отрендеренных изображениях.

Эксперименты

3.1. Постановка экспериментов

Оцениваются четыре варианта предложенного растрово-обоснованного метода извлечения, образованные сочетанием двух детекторов Grounding DINO — gdino-tiny и gdino-base — с двумя стратегиями построения целевой маски. Стратегия box-fill использует каждую обнаруженную ограничивающую рамку напрямую как заполненную прямоугольную маску, а стратегия sam-vit-base уточняет каждую обнаруженную рамку с помощью SAM перед фильтрацией примитивов. Все варианты используют одни и те же пороги: порог рамки 0,35, текстовый порог 0,25 и порог выбора примитива $\tau = 0{,}8$. Для каждой пары SVG–запрос предсказанные выборы объектов преобразуются в бинарные маски узлов и оцениваются согласно описанному протоколу. Приводятся как метрики на уровне векторов (точное совпадение, Node IoU и Node F1), так и вспомогательные визуальные метрики (PSNR и SSIM на области интереса переднего плана, ROI), причём метрики ROI предпочтительнее полнокадровых, поскольку многие целевые объекты занимают лишь малую часть холста SVG.

3.2. Результаты на полном бенчмарке

Результаты растрово-обоснованного метода на полной оценке бенчмарка приведены в таблице 2. Наиболее сильная конфигурация — gdino-base + box-fill — достигает лучших оценок среди протестированных вариантов; вариант gdino-tiny + box-fill близок к ней, что указывает на существенный вклад этапа фильтрации примитивов в итоговое качество. Примечательно, что уточнение SAM не улучшает извлечение векторных объектов в данной постановке: хотя SAM может производить более детальные растровые маски, эти маски не обязательно лучше для выбора примитивов SVG. В частности, примитив SVG часто следует сохранять, когда большая часть его отрендеренной опоры лежит внутри области объекта, даже если уточнённая маска не покрывает весь примитив; поэтому более грубые маски с заполнением рамки обеспечивают более сильное покрытие для фильтрации на уровне примитивов.

Таблица 2 - Результаты растрово-обоснованного извлечения на полной оценке бенчмарка

DOI: https://doi.org/10.60797/IRJ.2026.169.21.3

Детектор	Маскирование	Точное ↑	Node IoU ↑	Node F1 ↑	PSNR_ROI ↑	SSIM_ROI ↑
gdino-base	box-fill	0,130	0,348	0,413	20,98	0,647
gdino-base	sam-vit-base	0,107	0,305	0,364	17,93	0,617
gdino-tiny	box-fill	0,120	0,339	0,409	18,99	0,631
gdino-tiny	sam-vit-base	0,104	0,303	0,366	16,65	0,603

Примечание: метрики на уровне узлов оценивают выбор примитивов SVG, тогда как метрики ROI на отрендеренных изображениях сравнивают растеризованные предсказанное и эталонное извлечения внутри области переднего плана

3.3. Сравнение со структурированными базовыми методами на основе LLM

Растрово-обоснованный метод сравнивается со структурированными базовыми методами на основе LLM. Модели не просят генерировать SVG-фрагменты в свободной форме, поскольку это смешивало бы выбор объекта с синтезом SVG, ошибками форматирования и корректностью рендеринга. Вместо этого каждая модель получает индексированное представление SVG-документа и набор текстовых запросов и для каждого запроса возвращает бинарный массив, выровненный с индексами узлов SVG. Формат вывода обеспечивается с помощью JSON-схемы: для каждого запроса схема содержит обязательное поле маски, длина которой фиксирована и равна числу узлов SVG, а значения ограничены 0 или 1. Каждый ответ дополнительно проверяется на наличие всех требуемых масок и их ожидаемую длину; некорректные выходные данные отклоняются и запрашиваются повторно. Такая постановка даёт LLM прямой путь к формату вывода бенчмарка и делает сравнение более строгим, поскольку все методы оцениваются как предсказатели масок узлов по одним и тем же метрикам.

Оцениваются две модели через сервис OpenRouter: qwen/qwen3.6-35b-a3b и google/gemma-4-26b-a4b-it. Постановка с LLM ограничена длиной исходной разметки SVG: при ограничении входа в 32 тыс. символов только 173 из 268 SVG-сцен помещаются во входной бюджет LLM, а оставшиеся 95 сцен пропускаются при оценке LLM, тогда как растрово-обоснованный метод может обработать все SVG бенчмарка. Это подмножество следует интерпретировать как благоприятную постановку для LLM, поскольку более длинные SVG-документы исключаются до оценки. В таблице 3 все четыре растрово-обоснованные конфигурации сравниваются с базовыми методами на основе LLM на подмножестве, пригодном для LLM. Qwen конкурентоспособен по точному совпадению, но лучшая растрово-обоснованная конфигурация существенно сильнее по всем остальным метрикам, тогда как Gemma показывает существенно худшие результаты. Эти результаты предполагают, что современные LLM иногда могут идентифицировать компактные точные наборы узлов, но остаются менее надёжными для восстановления полной векторной структуры запрашиваемого объекта.

Таблица 3 - Сравнение со структурированными базовыми методами на основе LLM на подмножестве, пригодном для LLM

DOI: https://doi.org/10.60797/IRJ.2026.169.21.4

Метод	Точное ↑	Node IoU ↑	Node F1 ↑	PSNR_ROI ↑	SSIM_ROI ↑
gdino-base + box-fill	0,177	0,439	0,514	25,78	0,702
gdino-base + sam-vit-base	0,132	0,368	0,434	20,20	0,643
gdino-tiny + box-fill	0,160	0,426	0,507	22,68	0,684
gdino-tiny + sam-vit-base	0,126	0,367	0,438	18,54	0,633
Qwen 3.6 35B A3B	0,185	0,369	0,421	24,42	0,681
Gemma 4 26B A4B IT	0,054	0,102	0,120	10,61	0,421

Примечание: растрово-обоснованные варианты оцениваются на том же ограниченном по длине подмножестве SVG-сцен, что и Qwen

Сравнение также подчёркивает практические различия между двумя семействами методов. Растрово-обоснованный конвейер работает локально и потребовал менее 6 ГБ памяти GPU в экспериментах; он не требует помещения исходной разметки SVG в контекст языковой модели и поэтому может обработать все SVG-сцены бенчмарка. В отличие от него, базовые методы на основе LLM ограничены длиной входа: при ограничении в 32 тыс. символов 95 из 268 SVG-сцен пропускаются. Хотя структурированный вывод избегает разбора SVG в свободной форме, он всё же требует генерации, ограниченной схемой, и последующей проверки масок узлов точной длины. Эти ограничения делают современные LLM менее практичными как самостоятельные решения для точного доступа к сложным SVG-сценам на уровне объектов.

3.4. Качественный анализ

Отобранное качественное сравнение на одиннадцати примерах SVG–запрос представлено на рисунке 2. Каждая строка соответствует одной аннотированной цели в одной SVG-сцене, тогда как столбцы показывают входную сцену, эталонное извлечение, промежуточные выходы растрового обоснования и предсказания базовых методов на основе LLM. Такая компоновка позволяет исследовать оба этапа растрово-обоснованного конвейера: локализует ли Grounding DINO запрашиваемый объект и приводит ли полученная маска к разумному извлечению на уровне векторов.

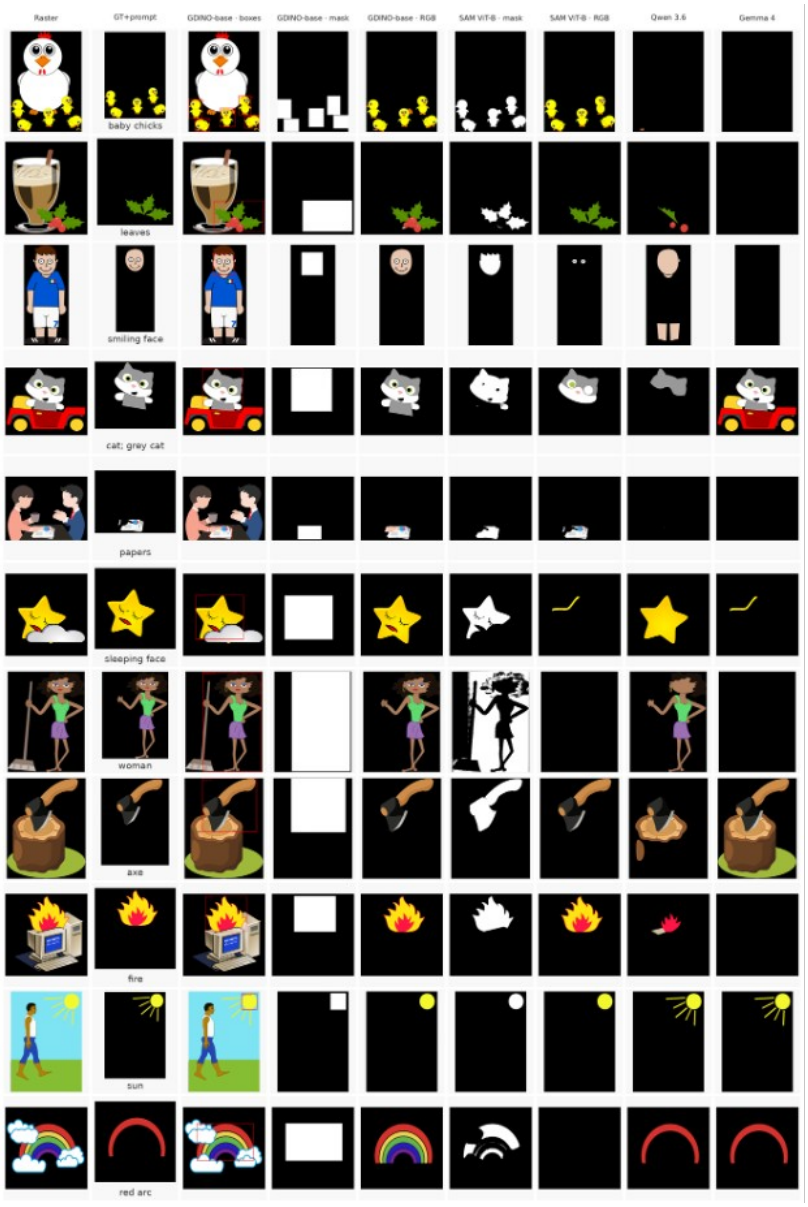

Рисунок 2 - Качественное сравнение на одиннадцати отобранных примерах SVG–запрос
DOI: https://doi.org/10.60797/IRJ.2026.169.21.5

Примечание: столбцы (слева направо): отрендеренная SVG-сцена, эталонное извлечение с меткой запроса, рамки Grounding DINO, маска и RGB-извлечение gdino-base + box-fill, маска и RGB-извлечение gdino-base + sam-vit-base, а также структурированные предсказания масок узлов от Qwen 3.6 и Gemma 4

Примеры выбраны для иллюстрации различных режимов бенчмарка. В нескольких случаях, таких как «птенцы», «листья», «улыбающееся лицо», «кот» и «бумаги», растрово-обоснованный метод существенно сильнее Qwen: это показывает, что предсказание масок узлов LLM может пропускать большие части объекта или выбирать структурно неправдоподобные элементы, даже когда запрашиваемый объект визуально ясен. В другой группе случаев, включая «звезду», «женщину», «топор» и «огонь», Qwen производит более осмысленные предсказания, но всё же остаётся ниже растрово-обоснованной базовой линии. Наконец, примеры «солнце» и «красная дуга» показывают случаи, где Qwen достигает лучшего перекрытия на уровне узлов, чем растрово-обоснованные варианты. Качественное сравнение подчёркивает два наблюдения: во-первых, сильная растровая локализация часто переходит в надёжный выбор примитивов SVG, особенно для объектов с компактной визуальной опорой; во-вторых, ни одно из семейств методов не решает задачу полностью — растрово-обоснованное извлечение может давать сбой, когда локализация пропускает части объекта или включает соседнее содержание, тогда как базовые методы на основе LLM остаются чувствительными к структуре SVG и точному предсказанию масок узлов.

Обсуждение и ограничения

Текущий метод ограничен качеством растрового обоснования. Если детектор пропускает часть запрашиваемого объекта, соответствующие примитивы SVG не могут быть восстановлены на этапе фильтрации; если обнаруженная область включает соседнее содержание, могут быть выбраны и не связанные с ним примитивы. Это особенно

проблематично для малых декоративных элементов, частично перекрывающихся объектов и объектов, чьи семантические границы нельзя чётко разделить в растровом пространстве. Протокол оценки также намеренно фокусируется на извлечении наборов объектов, а не на полном сопоставлении на уровне экземпляров: для каждого запроса предсказанные выборы объединяются в совмещённую маску узлов перед оцениванием, что делает оценку корректно определённой для запросов, таких как «облака» или «окна», но не измеряет, правильно ли метод разделяет отдельные экземпляры объектов. Разработка метрик уровня экземпляров для наборов узлов SVG является важным направлением будущей работы.

Результаты LLM предполагают, что универсальные языковые модели пока не являются полным решением для извлечения SVG-объектов. Даже когда задача упрощена до структурированного предсказания масок узлов, LLM остаются ограниченными длиной контекста, хрупкостью формата вывода и более слабыми оценками перекрытия на уровне узлов. Это указывает на необходимость в моделях, обученных или адаптированных специально для векторно-ориентированного обоснования, где супервизия связывает текст, растровый вид и примитивы SVG.

Заключение

В работе введена задача извлечения векторных объектов по текстовому запросу — восстановление редактируемых объектов из существующих SVG-сцен. Для её поддержки предложен бенчмарк с аннотациями на уровне узлов и оценкой, учитывающей векторную структуру, а также растрово-обоснованный метод извлечения, который локализует запрашиваемые объекты на отрендеренных изображениях, сопоставляет полученные области с отрисовываемыми примитивами SVG и реконструирует выбранные элементы в редактируемые под-SVG. Эксперименты показывают, что этот простой конвейер является сильной базовой линией для предложенного бенчмарка. Вариант с заполнением рамки превосходит уточнённые SAM маски при оценке на уровне узлов, что предполагает, что точные растровые границы не всегда оптимальны для восстановления полных векторных примитивов. Структурированные базовые методы на основе LLM остаются ограниченными длиной контекста и более слабыми оценками перекрытия на уровне узлов даже в ограниченной постановке предсказания масок узлов. В целом, работа предоставляет начальное определение задачи, бенчмарк, метод и эмпирический анализ для доступа к SVG-сценам на уровне объектов и поддерживает будущие творческие ИИ-системы, работающие не только с целыми изображениями, но и с редактируемыми векторными объектами и переиспользуемыми визуальными компонентами.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы на английском языке / References in English

1. Carlier A. DeepSVG: A Hierarchical Generative Network for Vector Graphics Animation / A. Carlier, M. Danelljan, A. Alahi [et al.] // Advances in Neural Information Processing Systems (NeurIPS). — 2020. — Vol. 33. — P. 16351–16361.

2. Li T.-M. Differentiable Vector Graphics Rasterization for Editing and Learning / T.-M. Li, M. Lukáč, M. Gharbi [et al.] // ACM Transactions on Graphics. — 2020. — Vol. 39. — № 6. — Art. 193. — P. 1–15.

3. Frans K. CLIPDraw: Exploring Text-to-Drawing Synthesis through Language-Image Encoders / K. Frans, L. Soros, O. Witkowski // Advances in Neural Information Processing Systems (NeurIPS). — 2022. — Vol. 35. — P. 5207–5218.

4. Jain A. VectorFusion: Text-to-SVG by Abstracting Pixel-Based Diffusion Models / A. Jain, A. Xie, P. Abbeel // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2023. — P. 1911–1920.

5. Xing X. DiffSketcher: Text Guided Vector Sketch Synthesis through Latent Diffusion Models / X. Xing [et al.] // Advances in Neural Information Processing Systems (NeurIPS). — 2023. — Vol. 36.

6. Wu R. IconShop: Text-Guided Vector Icon Synthesis with Autoregressive Transformers / R. Wu, W. Su, K. Ma [et al.] // ACM Transactions on Graphics. — 2023. — Vol. 42. — № 6. — Art. 208. — P. 1–14.

7. Wu R. Chat2SVG: Vector Graphics Generation with Large Language Models and Image Diffusion Models / R. Wu, W. Su, J. Liao // Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). — 2025. — P. 23690–23700.

8. Li J. UniSVG: A Unified Dataset for Vector Graphic Understanding and Generation with Multimodal Large Language Models / J. Li [et al.] // Proceedings of the ACM International Conference on Multimedia (ACM MM). — 2025. — arXiv:2508.07766.

9. Chen S. SVGenius: Benchmarking LLMs in SVG Understanding, Editing and Generation / S. Chen [et al.] // arXiv preprint arXiv:2506.03139. — 2025.

10. Rodriguez J.A. VectorGym: A Multitask Benchmark for SVG Code Generation, Sketching, and Editing / J.A. Rodriguez [et al.] // arXiv preprint arXiv:2603.29852. — 2026.

11. Nishina K. SVGEditBench V2: A Benchmark for Instruction-based SVG Editing / K. Nishina, Y. Matsui // arXiv preprint arXiv:2502.19453. — 2025.

12. Liu S. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection / S. Liu [et al.] // Proceedings of the European Conference on Computer Vision (ECCV). — 2024.

13. Kirillov A. Segment Anything / A. Kirillov [et al.] // Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). — 2023.

14. Xu H. SemLayer: Semantic-aware Generative Segmentation and Layer Construction for Abstract Icons / H. Xu [et al.] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2026. — arXiv:2603.24039.

15. Hu J. AmodalSVG: Amodal Image Vectorization via Semantic Layer Peeling / J. Hu [et al.] // arXiv preprint arXiv:2604.10940. — 2026.

16. Terral M. WildSVG: Towards Reliable SVG Generation Under Real-World Conditions / M. Terral [et al.] // arXiv preprint arXiv:2602.21416. — 2026.

17. Ren T. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks / T. Ren [et al.] // arXiv preprint arXiv:2401.14159. — 2024.